DOI:
10.1007/978-3-642-34222-6_9

Transparent Provenance Derivation for User Decisions

Ingrid Nunes[1,2], Yuhui Chen[2], Simon Miles[2], Michael Luck[2], and Carlos Lucena[1]

[1] LES, Departamento de Informática, PUC-Rio, Rio de Janeiro, Brazil
{ionunes,lucena}@inf.puc-rio.br
[2] Department of Informatics, King's College London, London, WC2R 2LS, United Kingdom
{yuhui.chen,simon.miles,michael.luck}@kcl.ac.uk

Abstract. It is rare for data's history to include computational processes alone. Even when software generates data, users ultimately decide to execute software procedures, choose their configuration and inputs, reconfigure, halt and restart processes, and so on. Understanding the provenance of data thus involves understanding the reasoning of users behind these decisions, but demanding that users explicitly document decisions could be intrusive if implemented naively, and impractical in some cases. In this paper, therefore, we explore an approach to transparently deriving the provenance of user decisions at query time. The user reasoning is simulated, and if the result of the simulation matches the documented decision, the simulation is taken to approximate the actual reasoning. The plausibility of this approach requires that the simulation mirror human decision-making, so we adopt an automated process explicitly modelled on human psychology. The provenance of the decision is modelled in Open Provenance Model (OPM), allowing it to be queried as part of a larger provenance graph, and an OPM profile is provided to allow consistent querying of provenance across user decisions.

Keywords: Decision making, explanation, OPM profile, inference.

1 Introduction

Humans are involved somewhere in most software processes, and the decisions they take are part of an explanation of the processes' effects. Therefore, as part of provenance information, it would be helpful to know the reasons why decisions were made as they were, including why a particular option was chosen and why others were not. While it is plausible to elicit something about a user's preferences over time, in many circumstances it is unrealistic to expect them to record the reasons behind every individual decision. If a decision is between many alternatives, each with pros and cons, and is influenced by a combination of different factors, it will not be apparent, just by knowing the user's preferences, why the decision was made. Moreover, a complex decision is influenced not just by what a user prefers, but also how they reason over the alternatives, i.e. psychological processes.

For example, when looking back at the total budget spent attending conferences by a group in a year, and considering how it might be reduced in subsequent years, it is relevant to consider why members of the group have chosen particular travel and accommodation options. The preferences of an individual may be apparent by looking across records from multiple years, but the choices made on a specific trip may be based on many attributes of the options available such as price, duration, location and facilities, and on preferences that do not consistently indicate one option, e.g. desire to spend little versus preference to share a hotel with a colleague with expensive tastes. The provenance of the budget spent can be seen as a process involving decisions drawing on many factors, and may be the result of heuristics that do not correspond exactly with 'rational' economic choices.

We wish to answer queries about the provenance of data where that provenance includes user decisions and the query relates to the reasons for those decisions. We are concerned with cases where the reasons for a decision are not immediately obvious as they require a choice between options with multiple attributes with pros and cons. We assume that, at recording time, the human reasoning is not captured, and instead derive a plausible explanation as part of the provenance query execution. This explanation is determined through simulating the user decision process using an automated decision making technique tailored to account for human psychological heuristics, e.g. preferring an option with uniformly acceptable attributes to one very good in some regard and very poor in another. The provenance of the simulated process is recorded. If the outcome of the decision-making process is the same as happened in reality, then the simulation provenance provides a plausible explanation of the user reasoning.

This problem is not one that has been tackled in depth in the literature, with notable exceptions. Naja et al. [10] consider a similar problem of the reasons behind decisions in a multi-agent simulation of an emergency response domain. They look at how the states a software agent transitioned through led to the decisions that were made. This is a comparable but not equivalent problem to our own. That is, they consider how the agent perceptions and prior actions influenced the decision rather than the reasoning on that decision itself. Moreover, they track the provenance of the software agent as developed for the response simulation, rather than trying to create a psychologically-realistic simulation of the decision reasoning. They construct an Open Provenance Model (OPM) [9] profile for the provenance, but this is specific to the emergency response domain rather than about decisions in general. Other work concerns the provenance of decisions, but again concern the gathering of data to inform the decision rather than the decision itself. For example, Kifor et al. [5] investigate the provenance of organ transplant decisions, but the decision itself is not modelled, only the observable factors used as input, while Missier et al. [8] record the quality of inputs to an automated decision, based on user criteria, to interpret the trustworthiness of the result. In the following sections, we first define the problem and provide a motivating example, before presenting the overall approach and its components: an *automated decision maker* and an *OPM profile for user decisions*, to later detail questions that can be answered regarding the human decisions.

2 Explaining User Decisions

We start by articulating the problem to be solved. Broadly, we aim to infer the provenance of user decisions, i.e. what reasoning led to those decisions, that take place within larger processes for which provenance is recorded. The decisions are choices between *options* based on criteria for making that decision, *preferences*. We assume some knowledge of those options and preferences in inferring the reasoning. The provenance is to be used to explain the effects of those decisions later in the larger processes.

2.1 Motivating Example

We will take a use case from the healthcare domain as our a running example.

Background. As part of the drug development process, clinical trials are conducted with patients by clinical researchers from, for example, pharmaceutical companies. Where the process of recruiting these patients has traditionally been carried out through personal meetings between researchers and doctors, automation is being brought to each stage. Projects such as Electronic Health Records for Clinical Research (EHR4CR) [3] or Translational Medicine and Patient Safety in Europe (TRANSFoRm) [2] aim to provide clinical research (CR) platforms that allow researchers to identify and recruit patients, querying their data from hospitals and other clinical data sites in multiple countries. The trial recruitment process is becoming one in which software processes are intermingled with human decisions (by researchers, patients, doctors, hospital auditors, etc.). Verified provenance data is critical in this context, due to the regulatory requirements applied to drug development and clinical trials. However, less strictly defined provenance information is also valuable in helping to refine trial recruitment. New clinical trials often have to face difficulties recruiting an adequate number of patients within a limited budget and timescale. A CR platform allows clinical researchers to design protocol feasibility studies with a set of patient eligibility criteria, send study queries to distributed clinical information systems, and rapidly get feedback on patient population numbers at each site and the geographic distribution of eligible patients. Understanding why a trial has not recruited enough patients means understanding what decisions were made during the studies and how.

Process. Alex is a clinical researcher with a pharmaceutical company. He is currently planning a clinical trial for a new drug that targets *Haemophilia A*. He needs to find sites for conducting the trial. He designs a study and composes a set of eligibility criteria for identifying suitable patients. For instance, he specifies inclusion criteria, such as "*male aged between 12 and 65*," "*immunocompetent with a CD4+ lymphocyte count* $> 200/mm^3$," and exclusion criteria, such as "*platelet count* $< 75{,}000/mm^3$." He submits the query to the a CR platform which in turn tries to discover eligible patients in the UK. After some time, the query result is ready, containing a list of feasible sites and important site-specific information, such as the number of eligible patients at the site, per-patient cost, and estimated local R&D approval time (Table 1).

Site	Number of eligible patients	Per-patient cost	Approximate local approval time (days)
A	30	£25,000	70
B	27	£22,000	60
C	22	£27,000	45

Table 1. CR query result example (illustrative only).

Decision. Alex decides which sites, if any, to recruit from. We assume that deciding to recruit patients from a site means that all eligible patients are recruited from that site, e.g. due to an agreement with sites to help them recoup admin costs. It is the provenance of this decision that we focus on.

Preferences. From past experience and the specification of an individual study, the researcher will have preferences on how to choose trial sites. For instance, if Alex needs at least 20 patients and accepts up to £600,000 trial cost and up to 80 days approval time, and is more concerned to reduce approval time than cost, then *C* is the ideal choice. If he instead prioritised number of patients recruited, *B* is preferable. *A* is discounted as it exceeds acceptable costs (30 patients x £25,000 = £750,000).

Options. There are eight options given the sites above: none (0), A only (A), B only (B), C only (C), A and B (AB), A and C (AC), B and C (BC), or all three (ABC).

2.2 Explanations

In order to justify a decision, different granularities of explanation can be given. High-level explanations either (i) highlight the positive and negatives aspects of chosen and rejected options [6, 7], giving arguments for or against options, or (ii) briefly indicate how the choice was made, as is typical in Recommender Systems (RSs) [17], e.g. "people who bought this product also bought..." However, for complex decisions, it can be unclear how the decision follows from the preferences known and options available. In such cases, more of the reasoning process must be exposed. Where option i was chosen over option j (amongst others), users ask questions such as the following.

– **Q1.** Are there preferences that compare i and j but did not affect the decision?
– **Q2.** Were any implicit (unstated) preferences considered?
– **Q3.** Do the positive aspects of i relative to j compensate its negative aspects?
– **Q4.** How much better is i to j relative to the trade-offs between i and other options?

3 Overall Approach and Background

In this section, we describe the components and methodology that comprise our approach, and provide a brief background on two works, which our approach is based on: the Open Provenance Model (OPM) [9] and a psychologically-inspired decision maker. The components required to realise our approach are the following.

System-independent provenance model. To form a connected account of provenance, including user decisions and software processes, we require a model that is system-independent. Here, we use the OPM.

Decision provenance pattern. We wish our solution to be generic and re-usable, allowing queries of a repeatable form over different decisions. Therefore, the provenance of a user decision should follow an application-independent pattern, expressed in this paper as a profile of OPM.

Human decision simulator. Most existing automated decision-makers do not attempt to reflect human decision making, but search for the choice that best matches the stated preferences. For our simulation, we use an existing decision making approach [14, 15] that explicitly applies heuristics observed in studies of human psychology.

Explanation from provenance queries. The results of the provenance recording and decision simulation should be a connected provenance graph. Finally, we need to provide some means to ask the provenance queries over this graph.

Our overall methodology is composed of six steps, detailed next.

1. As an application executes, an OPM graph is recorded documenting what has occurred in observable software processes.
2. A data item (OPM artifact) denotes a decision made by a user.
3. An automated decision-maker processes the known preferences potentially influencing the decision and set of options chosen between.
4. As the automated decision-maker executes, it documents its operations in OPM following a pre-defined profile for the provenance of a user decision.
5. If the outcome of the decision maker is the same as the actual decision, the graphs from steps 1 and 4 are combined to form a single graph.
6. Provenance queries that concern the reasons behind the decision can be executed.

3.1 Open Provenance Model

The Open Provenance Model (OPM) [9] is an abstract provenance model that describes past occurrences in terms of *artifacts*, immutable states of data items or physical objects, *processes*, actions performed on, using or generating artifacts, and *agents*, contextual entities acting as catalysts for processes. These entities are connected into graphs with edges from effect to cause, e.g. that a process used an artifact or an artifact was generated by a process. When depicted visually, as in Figures 1, 2 and 3, ovals denote artifacts and rectangles denote processes. An edge between an artifact and a process can include a *role* identifier, stating the artifact's function in the process, denoted by brackets after the edge type. Artifacts and processes can be *typed* by giving an annotation `opm:type=X`, where X is a unique type identifier.

To execute a query over an OPM graph, you need to know its structure. Ideally, queries can be re-used across similar applications, and so OPM *profiles* are used to give domain-specific extensions for OPM, allowing the graph structures to be common within that domain. An OPM profile is defined by *(i)* a unique global identifier; *(ii)* an optional controlled vocabulary for annotations; *(iii)* optional general guidance to express OPM graphs; and *(iv)* optional profile expansion rules. In the following sections, we describe the key elements of our method: the automated decision maker, and the OPM profile for user decisions. We then describe how the combined graph would be queried to answer questions about the reasons behind decisions.

3.2 Psychologically-inspired Automated Decision Making

The automated decision maker used to simulate the user decisions is described in prior work [14, 15]. As described in the published work, it has been evaluated to ensure it reflects the decisions that users would make given adequate information on the options. Here we summarise the key aspects, which are illustrated with a scenario in which a researcher is looking for an apartment to stay at, and each apartment is described in terms of the city zone that it is located, distance from university and price. The decision maker inputs are the user *preferences*, and the *options* available, specified in terms of their *attributes*. Derived from studies of how users express preferences in practice, there are seven kinds that can be specified, shown in Table 2. Preferences may apply only conditionally, where the condition is an expression in terms of attribute values. In addition, *priorities* can be expressed either between attributes or between preferences, so that the attribute/preference is given more weight in the decision making.

Two primary models are then constructed. The Preference Satisfaction Model (PSM) is a mapping of each attribute of each option to a rating of how much that option is individually desired, e.g. considering preferences 4 and 5, an apartment in zone 1 is mapped to *best*, while one in zone 2 to *prefer* (w.r.t. zone). The Options-Attribute Preference Model (OAPM) states, for each attribute of each option, how it compares to the same attribute of each other option, either better $(+)$, worse, $(-)$, similar $(\sim)$ or inconclusive (?), e.g. if Ap_A is cheaper than Ap_B then $OAPM[Ap\_A, Ap\_B, price] = +$. Where the explicitly stated preferences are insufficient for building these models, the decision maker will look for preferences *implied* by those stated. For example, if an upper bound is given as in preference 1, a goal to minimise this attribute is derived from it.

The relative benefits of options across all attributes are then calculated using preferences to derive *how much* an attribute value is better then another, and this cost-benefit analysis is combined with two principles from psychology on how humans make decisions [16]. The first, *extremeness aversion*, states that people avoid options that compromise one attribute too much to improve another. For example, an Ap_A is 2Km away from the university and costs £125 per week, Ap_B is 2.5Km away and costs £100, and Ap_C is 3Km away and costs £75. The costs of each option is compensated by its benefits, but people tend to choose Ap_B because its attributes are less extreme. The second, *trade-off contrast*, indicates that people consider the whole set of options when evaluating the trade-off between two options, i.e. the scale of differences across available options influences individual comparisons. Comparing only Ap_A and Ap_B, it is difficult to know if paying more £25 compensates being 0.5Km closer to the university,

Preference	Description	Example	#
Constraint	Specifies the values that attributes must (not) have	$uni < 4Km$	1
Goal	Specifies which attributes should be minimised or maximised	*minimise price*	2
Order	Specifies where one attribute value is preferred to another	$zone = 1 > zone = 2$	3
Qualifying Preference	States how much an attribute value is wanted or needed	*prefer* $zone = 1 \vee 2$	4
Rating Preference	Specifies which values are best or worst	$zone = 1$ *best*	5
Indifference	Specify where there is no preference between two attribute values	$zone = 1 \sim zone = 2$	6
Don't care	Specifies where an attribute is irrelevant to the decision	*don't care price*	7

Table 2. Preference types.

so people look at this relationship among all the other options to evaluate this particular one.

Many decision making systems have been proposed over the years, including Expert Systems (ESs), which capture domain knowledge to make decisions like a domain expert [11], Recommender Systems (RSs), which recommend options from a (huge) set based on statistical models [17], and Decision Support Systems (DSSs), which use decision making models, commonly inspired in economy [4], to make choices [6, 7]. For several use cases, it is important to explain how decisions made by these processes came about. For RS and DSS, explanations focus on indicating the general idea underlying the recommendation ("people that bought this product also bought...") or indicating positive and negative aspects of options. While enough in some situations, users sometimes need details to understand why and how an option should be chosen, and merely exposing the software process or its inputs may be not helpful. ESs typically present the chain of rules fired to produce a given output. This approach is limited by its specificity: rules are domain-specific and a huge amount of them are elicited for each ES, and thus there is no reuse across applications. As we will show in the next section, we present a generic OPM profile to try to capture the reasoning process enabling detailed questions to be answered.

4 An OPM Profile for Decision Making

As the decision is simulated by the above decision maker, it records the reasoning in OPM following a profile. The profile ensures consistency of OPM graphs for decision reasoning, so allowing reusable queries to be created. We refer to the profile as the User-Centric Preference-Based Decision (UCPB) profile. A base URI is used for all types defined, `http://www.les.inf.puc-rio.br/`, referred to with prefix `ucpb`. The profile's unique identifier is `ucbp:Profile`. The profile has all optional elements listed in Section 3 except for expansion rules.

The profile includes a graph template for the provenance of a decision, depicted in Figures 1 and 2 (split into parts for space reasons). Each artifact or process is given a URI type annotation, defined in Table 3, so that queries can identify what part of the reasoning process it represents. Where a subgraph is specific to one option and/or attribute, that subgraph will be repeated for each option and/or attribute considered, and the artifact/process type is shown as parametrised, e.g. `Extremeness(i)`.

Note that part of the provenance graph's value comes from connecting a decision with only those preferences that were taken into account, i.e. filtering for relevance. The provenance graph excludes preferences, priorities and weightings that did not influence the decision, and so a subset of those known of the user.

Figure 1 presents the part of the provenance graph that describes how an option was selected based on the decision values of options compared to the others. The decision making process finishes when an option i is selected from a set of options, based on the decision values of this option with respect to the others and vice-versa. A decision value, in turn, is the result of the weighted sum of the relative benefits between options, the trade-off contrast, and the extremeness aversion, the three human processes components that the technique simulates. Initially, individual attribute values are analysed

ApplyImplicitPreferences	Applies preferences implicitly derived from known user preferences.
AssessAttributeBenefit	Assesses the benefit of attribute a of option i w.r.t. option j.
AssessAttributeImportance	Builds a partial order of attributes, based on priorities.
AssessAvgTradeOff	Assesses the average of the cost-benefit relationship (trade-off) among all options.
AssessDistanceFromBest	Calculates the disadvantage of an option attribute w.r.t. the best possible value.
AssessExtremeness	Assesses option extremeness (standard deviation of the distance from best of each attribute).
AssessOptionAttribute	Assesses the preference for an option attribute value based on monadic preferences.
AssessOptionDecisionValue	Assesses a value that represents how an option is better than another.
Attribute	Criterion used to describe an option, which is associated with a attribute domain.
AttributeBenefit	Advantage (in percentage points) of the attribute value of option i w.r.t. option j.
AttributeDomain	Range of all possible values that can be assigned to an attribute.
AttributeFunction	WeightFunction parameterised to calculate attribute weights given an AttributePartialOrder.
AttributeIndifference	Priority that states that an attribute a is as important as attribute b.
AttributePartialOrder	Partial order among attributes, establishing an importance relationship.
AttributePriority	Priority that states that an attribute a is more important than attribute b.
AttributeWeight	Weight specified for an attribute, representing its importance.
AVPO	Partial order of values of a particular attribute, stands for attribute value partial order.
BuildAttributeValuePartialOrder	Builds a partial order of the values of an attribute, based on order preferences.
CalculateAttributeWeight	Calculates an attribute weight based on a function and the attribute importance.
CalculateFunctionParameters	Calculates the parameters of the WeightFunction based on the AttributePartialOrder.
CompareOptionsAttribute	Compares the attribute values of two options, establishing a preference order or indifference.
DecisionValue	Value (in percentage points) that represents how much an option is preferred w.r.t. another.
DistanceFromBest	Distance from an option attribute value (in percentage points) to the best possible value.
DontCare	Preference that specifies an attribute whose values are irrelevant for the decision.
EvaluateAllOptionBenefits	Evaluates the overall benefits of option i w.r.t. option j.
EvaluateExtremenessAversion	Evaluates the difference between the extremeness of two options.
EvaluateTradeOffContrast	Evaluates the difference between the trade-off of two options and the trade-off average.
Extremeness	Value that indicates (in percentage points) how extremeness an option is.
ExtremenessAversion	Value that indicates the benefit of an option for being less extreme than another.
Goal	Preference that states the desire of maximising or minimising an attribute value.
Indifference	Preference that indicates attribute values that are equally preferred.
ModifierScale	Scale that establishes a partial order of the strength of modifiers (performatives or rates).
MonadicPreference	Preference that refers to a single target, and evaluates it with modifiers.
OAPM	Options-attribute preference model, states preference between option attribute values.
OrderPreference	Preference that indicates that an attribute value is preferred to another.
PreferencePriority	Priority that states that a preference is preferred to another.
RelativeBenefit	Values that indicates (in percentage points) the advantage of an attribute value w.r.t. another.
PSM	Preference Satisfaction model, associates attribute values of options with a modifier.
SelectOption	Selects an option from those available based on decision values.
SelectedOption	Option selected from a set.
TradeOffContrast	Value that indicates the benefit of an option for having a good trade-off w.r.t another.
TO	Trade-off (cost-benefit relationship) between two options.
TOAvg	Average of the trade-offs among all options.
WeightEA	Weight of the extremeness aversion used in the decision function.
WeightTO	Weight of the trade-off contrast used in the decision function.
WeightFunction	Parameterised function (e.g. $f(x) = log_a x + b$) that is used to calculate attribute weights.

Table 3. Term definitions.

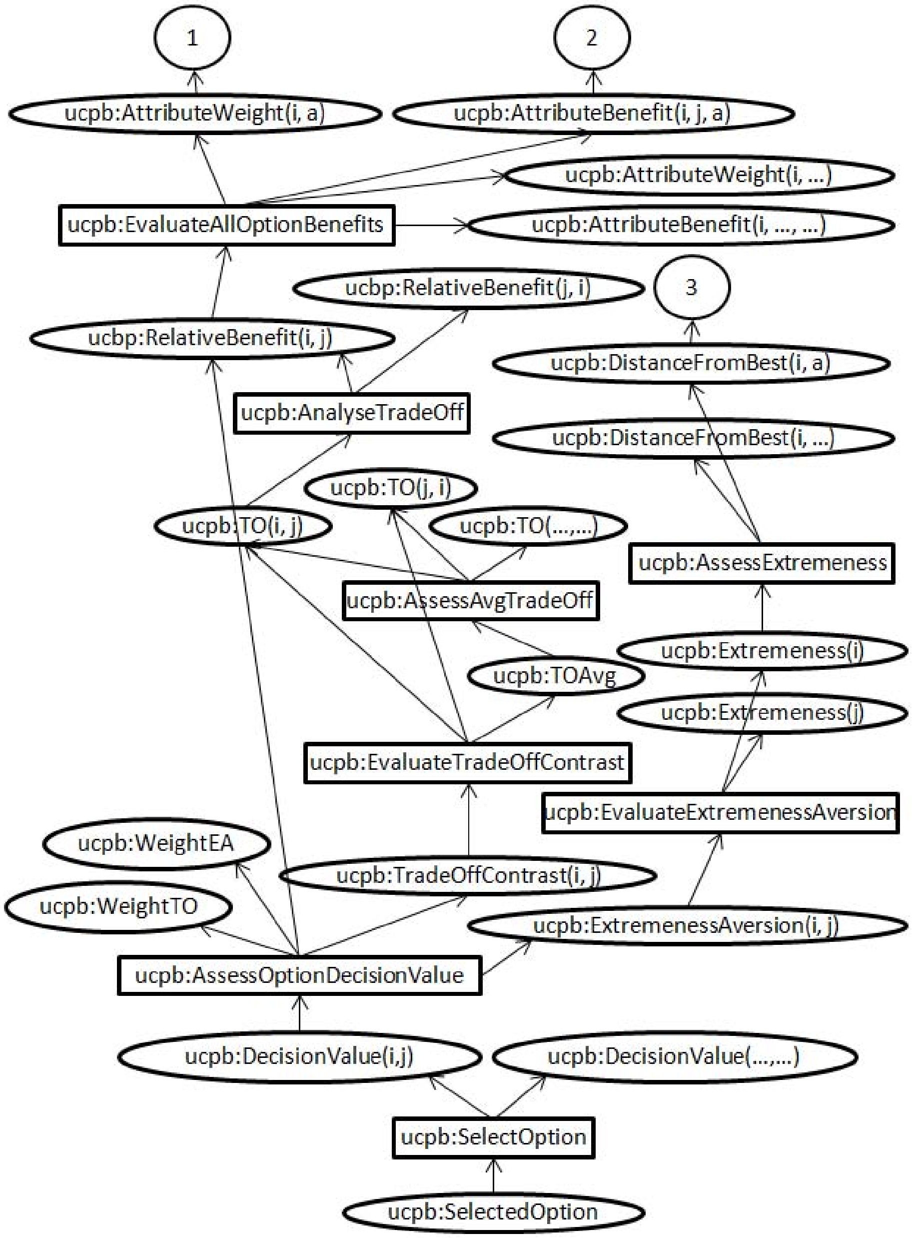

Fig. 1. Provenance graph I (prototype).

and their differences are evaluated. But, according to the importance of particular attributes, a small difference may be considered very significative. Then, people observe two other factors, which look at the relationship among attribute values. First, when an option compromise too much an attribute to compensate another, it is considered an extreme option, which is in general avoided by people (extremeness aversion). Second, as people often are not sure when a positive aspect of an option compensates a negative aspect, they look at this trade-off relationship of all options to make this evaluation, analysing the trade-off contrast. Options that have a good cost-benefit relationship are preferred. The trade-off contrast is calculated based on the benefit between an option with respect to another, which depends on two factors: (i) the weight of a particular attribute, which is specific for an option, detailed in Figure 2(1); and (ii) the benefit of a particular attribute, detailed in Figure 2(2). And the extremeness aversion compares how extreme options are, which is calculated as the standard deviation of the distances of an option attribute values to those of an option considered best — this is obtained from the provided preferences, and detailed in Figure 2(3).

Figure 2 details these three parts, whose leafs are preferences, or priorities in case of attribute weights. Therefore, by following a particular path of the tree beginning in the selected option one can understand the preference(s) that caused calculated values, which lead to the choice for that option. Different preferences are treated differently. Monadic preferences are first used to build the PSM, which in turn is used together with the remaining preferences to construct the OAPM. This model is later refined by considering implicit preferences in the explication process.

Returning to our example, as part of the scenario, Alex chose sites B and C (BC). We simulate the decision based on preferences we believe him to have. Specifically, Alex has the following goals: (P_1) maximise the number of eligible patients recruited; (P_2) minimise costs; (P_3) minimise approval time. He further has some qualifying preferences: (P_4) want around 50 patients; (P_5) accept spending between £1M and £1.2M. Finally, Alex has a priority: (P_6) prioritise number of patients over other attributes. Taking the eight options and the latter preferences, the decision maker simulates the decision, recording an OPM graph, an extract of which is shown in Figure 3.

5 Decision Provenance Queries

Given a provenance graph following our profile, queries can be made about the reasoning behind a decision. The following are examples, illustrated with our case study. They make reference to the chosen option i and another option j. To make the queries more precise, we will use an XPath-like notation, where each step in the path is the type of an artifact, process or edge, and a parent-child relation denotes that an edge links into or from an artifact or process. For example, `//ucpb:SelectOption/opm:used/*` returns all artifacts used by a ucpb:SelectOption process. The language is for illustration and is only semi-formal, but is similar to real provenance query languages [1].

Q1. Are there preferences that compare i and j but did not affect the decision? Alex chose option BC, recruiting 49 patients in total and not, for example, AB, recuiting 57. Querying the graph will tell us that BC was preferred to AB specifically with regard to

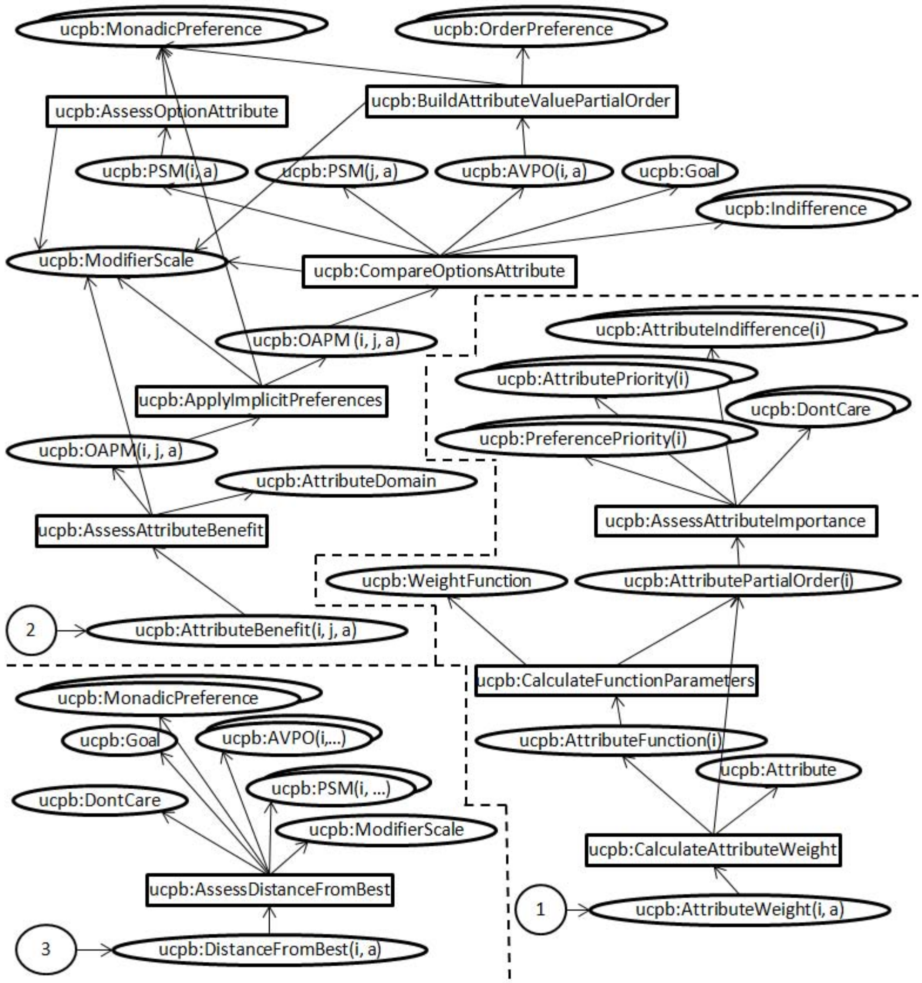

Fig. 2. Provenance graph II (prototype).

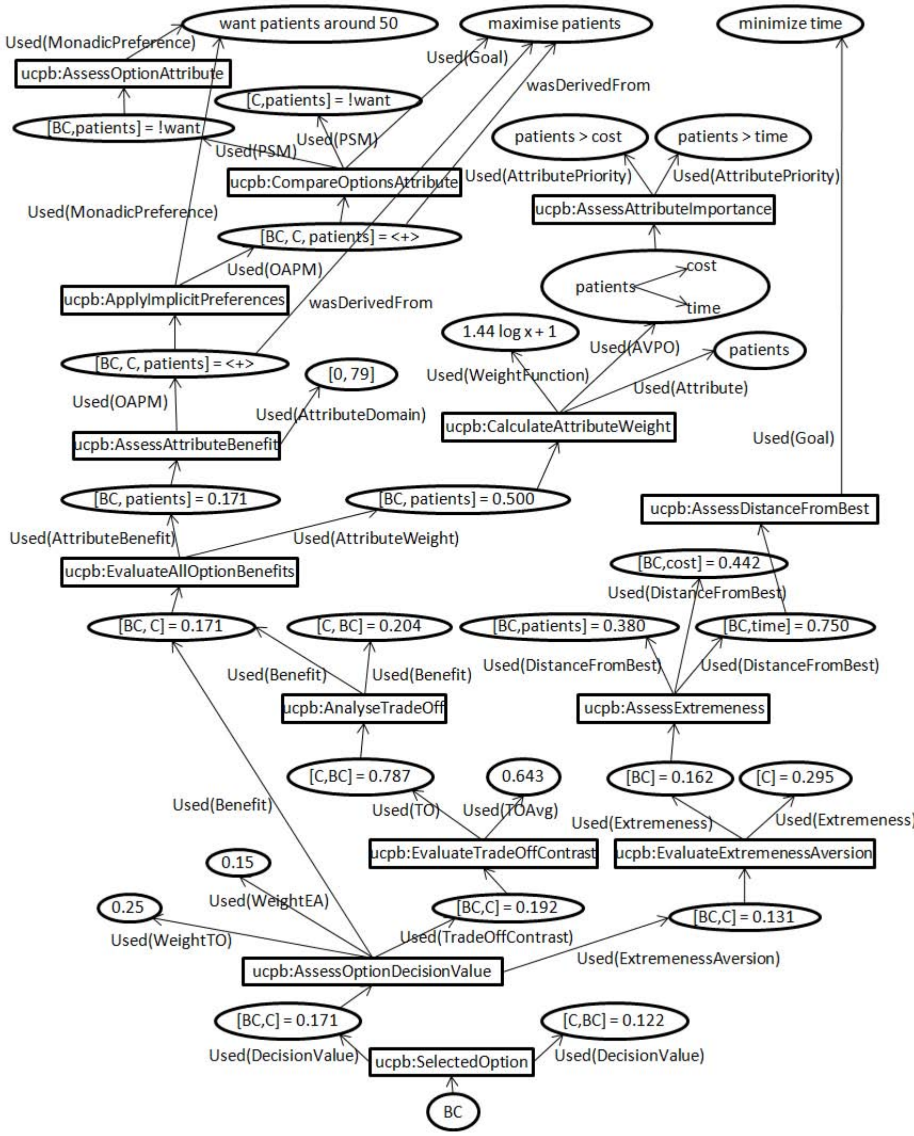

Fig. 3. CR Provenance Graph (partial).

the number of patients, recorded as artifact `ucpb:OAPM` having value "+". The graph further tells us the preference that was the reason for this decision, P_4, through a wasDerivedFrom OPM edge. Preference P_1 also concerns this attribute, and was an input to the process generating the `ucpb:OAPM`, but was not the reason, so no wasDerivedFrom edge exists. This query can be executed by first retrieving all preferences used in comparing options, `//ucpb:CompareOptionsAttribute/opm:used/*`, and then removing all those having a positive result on the chosen option `//ucpb:OAPM [opm:value='[BC]=+']/opm:wasDerivedFrom/*`. Those remaining cannot have influenced the final choice.

Q2. Were any implicit preferences considered? In our example, Alex had preference P_5 stating that a cost higher than £1M and lower than £1.2M is 'acceptable.' This is not necessarily a hard constraint, i.e. a cost lower or higher may still be a valid option, but values in that range fit in the 'acceptable' range of the modifier scale. This explicit preference also implies a further preference, P_7, that values outside of the interval are more acceptable if closer to it, e.g. £0.9M is closer to being acceptable than £0.8M. However, in this example, the explicitly stated preferences, including P_2, are adequate for making a comparison of options, so the implicit preference has no effect. This can be seen in the provenance graph where `ucpb:InferImplicitPreferences` does not alter the comparison value ('[B,C,patients]=<+>'), B is preferable to C with regards to number of patients, before and after the process). This query can be executed by detecting any instances where the OAPM output and input of the implicit preference step are different, `//ucpb:OAPM[opm:value!=opm:wasGeneratedBy/ //ucpb:Apply ImplicitPreferences/opm:used/ ucpb:OAPM/opm:value]`.

Q3. Do the positive aspects of option i relative to option j compensate its negative aspects? Alex's decision making was not necessarily a pure reflection of the positive and negative attribute differences, but will have been influenced by human psychological processes. Here we model two known effects described earlier, trade-off evaluation and extremeness aversion. The influence of these on the eventual decision is weighted in the simulation (0.25 and 0.15 respectively). The weights may be derived from observations of Alex, or based on averages across a wider population. We can then ask whether these psychological processes affected the particular decision. In this case, the benefit of option BC over C, given value 0.171, was lower than the benefit of C over BC (given value 0.204, shown in Figure 3 as an artifact used by `ucpb:AnalyseTradeOff`). However, the difference between the options is small relative to the differences across the set (trade-off) and C has a higher rating primarily because one attribute, approval time, is very good but at the expense of another, eligible patients (extremeness aversion). Because of this, the eventual decision chooses BC. This query can be executed by checking whether the benefits comparison without trade-off and extremeness aversion, `//opm:wasGeneratedBy/ucpb:EvaluateAllOptionBenefits`, results in one option being preferred to another while the inputs to the final selection `//ucpb:SelectedOption/opm:used` show the reverse.

Q4. How much better is i to j relative to the trade-offs between i and other options? The trade-off between two options is always evaluated based on its comparison with other trade-offs, i.e. what is taken into account is the trade-off *contrast*. Therefore, in order to understand the trade-off between two options it is important to allow users to verify the average of the trade-offs. In Figure 3, it can be observed that the trade-off (cost-benefit ratio) between options site C only and sites B and C together is 0.787, which is higher (worse) than the average 0.643. This query can by comparing the particular trade-off measures, `//opm:wasGeneratedBy/ucpb:AnalyseTradeOff`, with the average of those values.

Answers to Q1 and Q2 will refer to attributes, leading to further questions such as "Why was i's value for attribute a considered better than j's?" or "Why was attribute a more important than b?"

Recording the decision making process that matches a user choice using our profile also allows generating high-level explanations, mentioned before. Even though they may not be enough in some cases, mainly when there is the need for a detailed explanation between pros and cons of individual attributes, which is the case of the queries above, giving an initial high-level explanation may be useful. However, in order to do so, existing approaches [6, 7] need as input attribute weights and values, and therefore a provenance graph is also needed to generate such explanations. We are currently working on this direction as well. In order to identify the explanations users expect to receive as a justification for a choice, we have performed a study on how people justify their choices [12, 13], assuming that given explanations are those they expect to receive. As a result of this study, we derived guidelines and patterns of explanations, providing guidance on explanations to be generated and the context in which each of them should be provided. Given this study, we are developing a technique that generates explanations following derived patterns, taking as inputs a provenance graph built using our proposed profile.

6 Conclusion

Knowing the reasoning of decisions taken by humans in the context of partially automated systems is crucial in many domains, such as those that involve design decisions: clinical trials, software development and civil construction. Nevertheless, it is unrealistic to expect that all decisions are justified by users given the time and effort that this activity requires. Therefore, we presented in this paper an approach that aims at automating the process of recording humans decisions.

Our approach consists of observing user's choices in the context of software applications that support people to manage tasks that involve decision making. With an automated decision-maker whose goal is to simulate human reasoning, we detect situations in which a human choice matches that of the decision-maker, whose reasoning process can be used to justify a human decision. In order to record such explanations, we based our approach on the Open Provenance Model (OPM), which is a generic model to represent the provenance of data (or physical objects, ...) and is being adopted as a pattern to allow the interoperability of systems. We proposed an OPM profile, which is an extension of this generic model to accommodate the specific artifacts of the automated decision-maker, and the processes associated with it. Moreover, we showed how to query provenance graphs built with our profile in order to obtain explanations to justify choices made based on preferences. Human decision making is very complex, and therefore there are many decisions that our human decision simulator still cannot reproduce. Our future work is thus to incorporate to our automated decision-maker other principles of psychology that can help to explain human decisions.

References

1. Anand, M.K., Bowers, S., Altintas, I., Ludaescher, B.: Approaches for exploring and querying scientific workflow provenance graphs. In: IPAW'10 (2010)
2. Delaney, B., Taweel, A., et al: Transform:translational medicine and patient safety in europe. In: Proceedings of the AMIA 2010 Annual Symposium (2010)
3. Kalra, D., Schmidt, A., Potts, H.W.W., Dupont, D., Sundgren, M., De Moor, G.: Case report from the ehr4cr project— a european survey on electronic health records systems for clinical research. iHealth Connections (2011)
4. Keeney, R.L., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs. John Wiley & Sons, Inc, New York (1976)
5. Kifor, T., Varga, L., Vazquez-Salceda, J., Alvarez, S., Willmott, S., Miles, S., Moreau, L.: Provenance in agent-mediated healthcare systems. IEEE Intelligent Systems 21(6), 38–46
6. Klein, D.A., Shortliffe, E.H.: A framework for explaining decision-theoretic advice. Artif. Intell. 67, 201–243 (June 1994)
7. Labreuche, C.: A general framework for explaining the results of a multi-attribute preference model. Artif. Intell. 175, 1410–1448 (May 2011)
8. Missier, P., Embury, S., Stapenhurst, R.: Exploiting provenance to make sense of automated decisions in scientific workflows. In: IPAW 2008 (2008)
9. Moreau, L., Clifford, B., Freire, J., Futrelle, J., Gil, Y., Groth, P., Kwasnikowska, N., Miles, S., Missier, P., Myers, J., Plale, B., Simmhan, Y., Stephan, E., den Bussche, J.V.: The Open Provenance Model core specification (v1.1). Future Gener. Comput. Syst. 27(6), 743–756 (June 2011)
10. Naja, I., Moreau, L., Rogers, A.: Provenance of decisions in emergency response environments. In: IPAW 2010. Springer-Verlag, Troy, NY, USA (2010)
11. Nakatsu, R.T.: Explanatory power of intelligent systems. In: Gupta, J.N.D., Forgionne, G.A., Mora T., M. (eds.) Intelligent Decision-making Support Systems, pp. 123–143. Decision Engineering, Springer London (2006)
12. Nunes, I., Miles, S., Luck, M., Lucena, C.: Investigating explanations to justify choice. In: 20th International Conference on User Modeling, Adaptation and Personalization (UMAP 2012). Canada (2012)
13. Nunes, I., Miles, S., Luck, M., Lucena, C.: A study on justifications for choices: Explanation patterns and guidelines. Tech. Report CS-2012-03, University of Waterloo, Canada (2012)
14. Nunes, I., Miles, S., Luck, M., Lucena, C.: User-centric preference-based decision making. In: AAMAS'12 (2012), (to appear)
15. Nunes, I., Miles, S., Luck, M., Lucena, C.: User-centric principles in automated decision making. In: 21st Brazilian Symposium on Artificial Intelligence (SBIA 2012) (2012), (to appear)
16. Simonson, I., Tversky, A.: Choice in context: Tradeoff contrast and extremeness aversion. Journal of Marketing Research 29(3), 281–295 (1992)
17. Tintarev, N., Masthoff, J.: A survey of explanations in recommender systems. In: 23rd International Conference on Data Engineering Workshop. pp. 801–810. IEEE (2007)